

LSI-3123

Quadrature Encoder Counter Card

Software Manual (V1.1)

健昇科技股份有限公司

JS AUTOMATION CORP.

台北縣汐止市中興路 100 號 6 樓

6F, No. 100, Chungshin Rd.

Shitsu, Taipei, Taiwan, R.O.C.

TEL : 886-2-2647-6936

FAX : 886-2-2647-6940

<http://www.automation.com.tw>

E-mail : control.cards@automation.com.tw

Correction record

Version	Record
1.1	1. Re-alignment of text

Contents

1.	How to install the software of LSI3123	4
1.1	Install the PCI driver.....	4
1.2	How to install the utilities.....	4
2.	About the LSI-3123 software.....	5
2.1	What you need to get started.....	5
2.2	Software programming choices	5
3.	LSI3123 Language support	6
3.1	Building applications with the LSI3123 software library.....	6
3.2	LSI3123 Windows libraries	6
4.	Software overview	7
4.1	Initialization	7
4.2	I/O Port R/W.....	7
4.3	Homing functions	7
4.4	Counter functions.....	7
4.5	Security functions	8
4.6	Miscellaneous functions	8
4.7	Error conditions	8
5.	Flow chart of implement an application.....	9
5.1	LSI3123 Flow chart of implementation.....	9
6.	Function reference.....	11
6.1	Error codes and CardID	11
6.2	Variable data types	12
6.3	Programming language considerations	13
6.4	LSI3123 Functions.....	15
	Initialization function group	15
	lsi3123_initial	15
	lsi3123_close	15
	lsi3123_info	15
	Input/output function group.....	16
	lsi3123_set_input_polarity	16
	lsi3123_set_led_polarity.....	16
	lsi3123_read_input_polarity	17
	lsi3123_read_input_status	17
	Homing function group.....	18
	lsi3123_set_hard_homing.....	18
	lsi3123_read_hard_homing_flag	18
	lsi3123_soft_homing_command.....	18
	Counter function group.....	19
	lsi3123_set_counter_mode	19

lsi3123_set_quadrature_times	19
lsi3123_read_counter.....	19
lsi3123_load_counter.....	20
lsi3123_latch_control	20
lsi3123_latch_mode.....	20
lsi3123_preset_counter	21
lsi3123_read_latch_flag.....	21
lsi3123_read_latch_value	21
Security function group	22
lsi3123_set_password	22
lsi3123_change_password	22
lsi3123_clear_password.....	23
lsi3123_read_security_status	23
lsi3123_unlock_security	24
Miscellaneous function group.....	25
lsi3123_read_parameters	25
6.5 Dll list	26
7. LSI-3123 Error codes summary	27
7.1 LSI3123 Error codes table	27

1. **How to install the software of LSI3123**

1.1 Install the PCI driver

The PCI card is a plug and play card, once you add a new card on the window system will detect while it is booting. Please follow the following steps to install your new card.

In Windows 98/2000 system you should: (take win98 as example)

1. Make sure the power is off
2. Plug in the interface card
3. Power on
4. A hardware install wizard will appear and tell you it finds a new PCI card
5. Tell the wizard the directory of the driver files (\LSI3123\win98\driver), then it will automatically setup the driver
6. After installation, power off
7. Power on , it's ready to use

1.2 How to install the utilities

Execute the file \install\setup3123.exe will automatically install all the utilities and setup a new directory for your new interface card.

For a more detail descriptions, please visit our user club

<http://automation.com.tw> and register as a member then download the file "Installation" which will take you go through step by step.

2. **About the LSI-3123 software**

LSI3123 software includes a set of dynamic link library (DLL) and system driver that you can utilize to control the I/O card's ports and points separately.

Your LSI3123 software package includes setup driver, tutorial example and test program that help you how to setup and run appropriately, as well as an executable file which you can use to test each of the LSI3123 functions within Windows' operation system environment.

2.1 What you need to get started

To set up and use your LSI3123 software, you need the following:

- LSI3123 software
- LSI3123 hardware
 - Main board
 - Wiring board (Option)

2.2 Software programming choices

You have several options to choose from when you are programming LSI3123 software. You can use Borland C/C++, Microsoft Visual C/C++, Microsoft Visual Basic, or any other Windows-based compiler that can call into Windows dynamic link libraries (DLLs) for use with the LSI3123 software.

3. **LSI3123 Language support**

The LSI3123 software library is a DLL used with Windows 98/2000/XP. You can use these DLL with any Windows integrating development environment that can call Windows DLLs.

3.1 Building applications with the LSI3123 software library

The LSI3123 function reference topic contains general information about building LSI3123 applications, describes the nature of the LSI3123 files used in building LSI3123 applications, and explains the basics of making applications using the following tools:

Applications tools

- ◆ **Borland C/C++**
- ◆ **Microsoft Visual C/C++**
- ◆ **Microsoft Visual Basic**

If you are not using one of the tools listed, consult your development tool reference manual for details on creating applications that call DLLs.

3.2 LSI3123 Windows libraries

The LSI3123 for Windows function library is a DLL called **LSI3123.dll**. Since a DLL is used, LSI3123 functions are not linked into the executable files of applications. Only the information about the LSI3123 functions in the LSI3123 import libraries is stored in the executable files.

Import libraries contain information about their DLL-exported functions. They indicate the presence and location of the DLL routines. Depending on the development tools you are using, you can make your compiler and linker aware of the DLL functions through import libraries or through function declarations.

Refer to **Table 1** to determine to which files you need to link and which to include in your development to use the LSI3123 functions in LSI3123.dll.

Header Files and Import Libraries for Different Development Environments		
Development Environment	Header File	Import Library
Microsoft C/C++	lsi3123.h	lsi 3123VC.lib
Borland C/C++	lsi 3123.h	lsi 3123BC.lib
Microsoft Visual Basic	lsi 3123.bas	

Table 1

4. Software overview

These topics describe the features and functionality of the LSI3123 boards and briefly describes the LSI3123 functions.

4.1 Initialization

You need to initialize each time you run your application and close your driver before you will close applications.

[*lsi3123 initial\(\)*](#)

[*lsi3123 close\(\)*](#)

If you need to get the card address assigned by the system,

[*lsi3123 info\(\)*](#) will do.

4.2 I/O Port R/W

Use the following functions for I/O port output value reading and control:

Set I/O port polarity and input phase polarity use:

[*lsi3123 set input polarity\(\)*](#)

[*lsi3123 set led polarity\(\)*](#)

To read back input polarity use:

[*lsi3123 read input polarity\(\)*](#) and to read input status with

[*lsi3123 read input status\(\)*](#).

4.3 Homing functions

If you want to clear counter with external trigger , use

[*lsi3123 set hard homing \(\)*](#) first to setup external trigger function. After configuration, use

[*lsi3123 read hard homing flag \(\)*](#) to check the occurrence of external trigger.

Sometimes you want to clear counter with software, use

[*lsi3123 soft homing command\(\)*](#) to implement.

4.4 Counter functions

Counting is the major function of this card. To use counter function setup the function mode is the first step. Use

[*lsi3123 set counter mode\(\)*](#) to configure the working mode of the counter.

If you are working with a quadrature encoder use

[*lsi3123 set quadrature times\(\)*](#) to setup the multiple rate.

To read the counter on the fly or to load it

[lsi3123_read_counter\(\)](#) and [lsi3123_load_counter\(\)](#) will do.

If you are using an external device trigger to load the counter, first you must set the latch control mode.

[lsi3123_latch_control\(\)](#) is required at initial stage to configure the trigger source and [lsi3123_latch_mode\(\)](#) to setup the trigger mode and finally the value to be load should set by [lsi3123_preset_counter\(\)](#).

If you are using an external device trigger to latch the counter on the fly, first you must set the latch control mode.

[lsi3123_latch_control\(\)](#) is required at initial stage to configure the trigger source and [lsi3123_latch_mode\(\)](#) to setup the trigger mode and verify if any trigger occurs by [lsi3123_read_latch_flag\(\)](#). If it does read the latched value by using [lsi3123_read_latched_value\(\)](#).

4.5 Security functions

Since LSI3123 is a general purpose card, anyone who can buy from JS automation corp. or her distributors, your program is the fruit of your intelligence, un-authorized copy maybe prevent by the security function enabled.

You can use

[lsi3123_set_password\(\)](#) to set password and [lsi3123_change_password\(\)](#) to change it.

If you don't want to use security function after the password being setup,

[lsi3123_clear_password\(\)](#) will reset to the virgin state.

Once the password is set, any function call of the dll's (except for the security functions) will be blocked until the

[lsi3123_unlock_security\(\)](#) unlock the security.

You can also use [lsi3123_read_security_status\(\)](#) to check the current status of security.

4.6 Miscellaneous functions

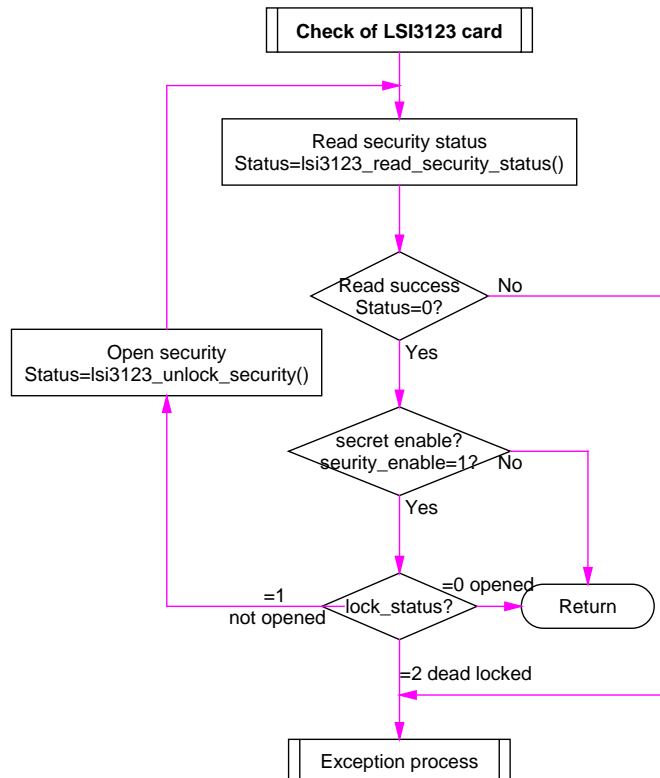
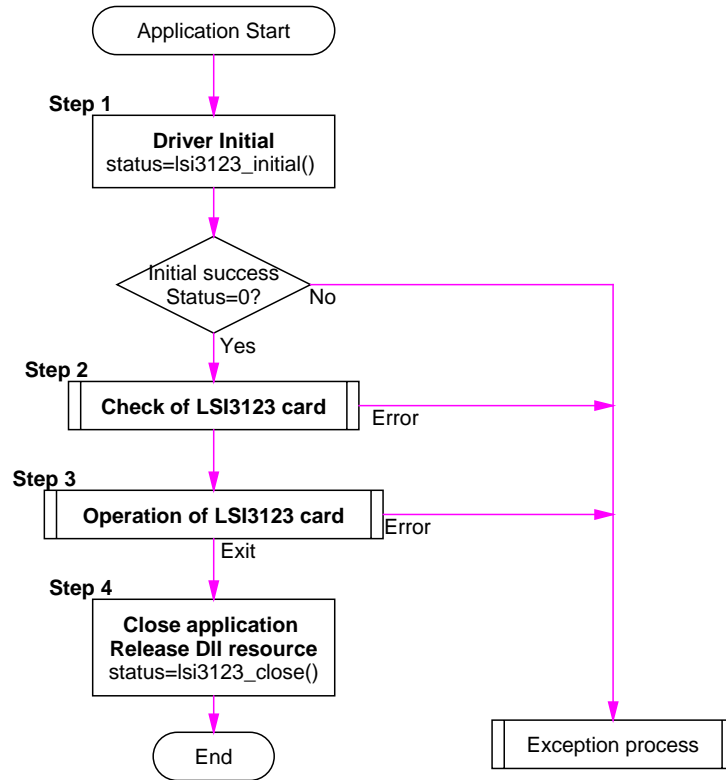
Sometimes you need to check the working modes, use [lsi3123_read_parameters\(\)](#) to check the homing mode, counter mode, latch mode, multiple rate and latch control.

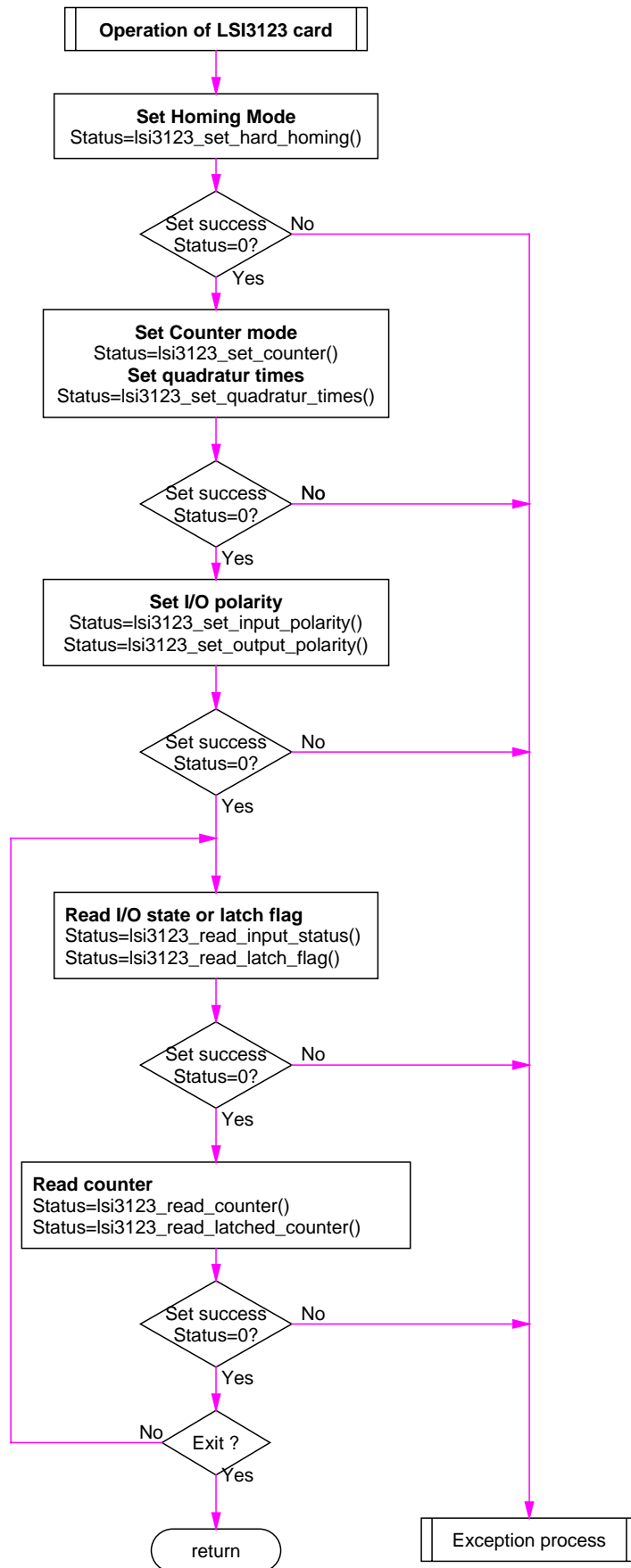
4.7 Error conditions

LSI3123 dll's will report several errors according the conditions it occurs. Error Codes contains a detailed listing of the error status returned by LSI3123 functions.

5. Flow chart of implement an application

5.1 LSI3123 Flow chart of implementation





6. **Function reference**

These topics contain detailed descriptions of each LSI3123 function. The functions are arranged by function group. Refer to LSI3123 Function Reference for additional information.

6.1 Error codes and CardID

Every LSI3123 function is consist of the following format:

Status = function_name (parameter 1, parameter 2, ... parameter n)

Each function returns a value in the **Status** global variable that indicates the success or failure of the function. A returned **Status** equal to zero that indicates the function executed successfully. A non-zero status indicates failure that the function did not execute successfully because of an error, or executed with an error.

Note : **Status** is a 32-bit unsigned integer.

The first parameter to almost every LSI3123 function is the parameter **CardID** which is located the driver of LSI3123 board you want to use those given operation. The **CardID** is assigned by ROTARY(or DIP) SW. You can utilize multiple devices with different card CardID within one application; to do so, simply pass the appropriate **CardID** to each function.

Note: **CardID** is set by ROTARY SW (**0x0-0xF**)

6.2 Variable data types

Every function description has a parameter table that lists the data types for each parameter. The following sections describe the notation used in those parameter tables and throughout the manual for variable data types.

Primary Type Names					
Name	Description	Range	C/C++	Visual BASIC	Pascal (Borland Delphi)
u8	8-bit ASCII character	0 to 255	char	Not supported by BASIC. For functions that require character arrays, use string types instead.	Byte
I16	16-bit signed integer	-32,768 to 32,767	short	Integer (for example: deviceNum%)	SmallInt
U16	16-bit unsigned integer	0 to 65,535	unsigned short for 32-bit compilers	Not supported by BASIC. For functions that require unsigned integers, use the signed integer type instead. See the i16 description.	Word
I32	32-bit signed integer	-2,147,483,648 to 2,147,483,647	long	Long (for example: count&)	LongInt
U32	32-bit unsigned integer	0 to 4,294,967,295	unsigned long	Not supported by BASIC. For functions that require unsigned long integers, use the signed long integer type instead. See the i32 description.	Cardinal (in 32-bit operating systems). Refer to the i32 description.
F32	32-bit single-precision floating-point value	-3.402823E+38 to 3.402823E+38	float	Single (for example: num!)	Single
F64	64-bit double-precision floating-point value	-1.797683134862315E+308 to 1.797683134862315E+308	double	Double (for example: voltage Number)	Double

Table 2

6.3 Programming language considerations

Apart from the data type differences, there are a few language-dependent considerations you need to be aware of when you use the LSI3123 API. Read the following sections that apply to your programming language.

Note: Be sure to include the declaration functions of LSI3123 prototypes by including the appropriate LSI3123 header file in your source code. Refer to Building Applications with the LSI3123 Software Library for the header file appropriate to your compiler.

6.3.1 C/C++

For C or C++ programmers, parameters listed as Input/Output parameters or Output parameters are pass-by-reference parameters, which means a pointer points to the destination variable should be passed into the function. For example, the Read Port function has the following format:

```
Status = lsi3123_read_port(CardID, port, data);
```

where **CardID** and **port** are input parameters, and **data** is an output parameter. Consider the following example:

```
u8 CardID, port;  
u8 data,  
u32 Status;  
Status = read_port (CardID, port, &data);
```

6.3.2 Visual basic

The file LSI3123.bas contains definitions for constants required for obtaining DIO Card information and declared functions and variable as global variables. You should use these constants symbols in the LSI3123.bas, do not use the numerical values.

In Visual Basic, you can add the entire LSI3123.bas file into your project. Then you can use any of the constants defined in this file and call these constants in any module of your program. To add the LSI3123.bas file for your project in Visual Basic 4.0, go to the **File** menu and select the **Add File...** option. Select LSI3123.bas, which is browsed in the LSI3123 \ api directory. Then, select **Open** to add the file to the project.

To add the LSI3123.bas file to your project in Visual Basic 5.0 and 6.0, go to the **Project** menu and select **Add Module**. Click on the Existing tab page. **Select** LSI3123.bas, which is in the LSI3123 \ api directory. Then, select **Open** to add the file to the project.

6.3.3 Borland C++ builder

To use Borland C++ builder as development tool, you should generate a .lib file from the .dll file by implib.exe.

implib LSI3123bc.lib LSI3123.dll

Then add the **LSI3123bc.lib** to your project and add

#include "LSI3123.h" to main program.

Now you may use the dll functions in your program. For example, the Read Port function has the following format:

Status = lsi3123_read_input_status(CardID, port, data);

where **CardID** and **port** are input parameters, and **data** is an output parameter. Consider the following example:

u16 CardID, port;

u8 data;

u32 Status;

Status = read_port (CardID, port, &data);

Initialization function group

● **lsi3123_initial**

Format : u32 Status =lsi3123_initial (void)

Purpose: Initialize the lsi3123 resource when start the Windows applications.

● **lsi3123_close**

Format : u32 Status =lsi3123_close (void);

Purpose: Release the lsi3123 resource when close the Windows applications.

● **lsi3123_info**

Format : u32 Status =lsi3123_info (u8 CardID,u16 *address);

Purpose: To access the lsi3123 physical address assigned by OS

Input:

Name	Type	Description
CardID	u8	assigned by DIP(rotary) SW

Output:

Name	Type	Description
address	u16	physical address assigned by OS

Input/output function group

● lsi3123 set input polarity

Format : u32 Status = lsi3123_set_input_polarity(u8 CardID,u8 Axis,u8 point,u8 polarity)

Purpose: Set input polarity

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
point	u8	0: A phase input 1: B phase input 2: C(Zero) phase input 4: latch input while Axis=0, latch input is Probe_IN while Axis=1, latch input is push botton input (Foot_SW_IN)
Polarity	u8	0: normal 1: invert

● lsi3123 set led polarity

Format : u32 Status = lsi3123_set_led_polarity(u8 CardID,u8 polarity)

Purpose: Set the LED polarity. The LED is activated by Probe_IN(LatchIN0).

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
polarity	u8	0: normal 1: inverse

- **lsi3123 read input polarity**

Format : u32 Status = lsi3123_read_input_polarity(u8 CardID,u8 Axis,u8 point,
u8 *polarity)

Purpose: Read input polarity

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
point	u8	0: A phase input 1: B phase input 2: C(Zero) phase input 4: latch input while Axis=0, latch input is Probe_IN while Axis=1, latch input is push botton input (Foot_SW_IN)

Output:

Name	Type	Description
Polarity	u8	0: normal 1: invert

- **lsi3123 read input status**

Format : u32 Status = lsi3123_read_input_status(u8 CardID,u8 Axis,u8 point,u8 *state)

Purpose: Read the input status

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
point	u8	0: A phase input 1: B phase input 2: C(Zero) phase input 4: latch input while Axis=0, latch input is Probe_IN while Axis=1, latch input is push botton input (Foot_SW_IN)

Output:

Name	Type	Description
state	u8	input status

Homing function group

● lsi3123 set hard homing

Format : u32 Status = lsi3123_set_hard_homing(u8 CardID,u8 Axis,u8 mode)

Purpose: To set the working mode of clearing counter.

Input:

Name	Type	Description
CardID	u8	assigned by DIP(rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
mode	u8	00h: no hardware homing (default) 01h: clear counter at A,B,C(Zero) phases goes low simultaneously 05h: clear counter at C(Zero) phases goes low 06h: clear counter at A,B, C (Zero) phases goes low simultaneously and reset to “no hardware homing” after the occurrence 0Ah: clear counter at C(Zero) phases goes low and reset to “no hardware homing” after the occurrence

● lsi3123 read hard homing flag

Format : u32 Status = lsi3123_read_hard_homing_flag(u8 CardID,u8 Axis,u8 *flag)

Purpose: Read hard homing flag for checking the occurrence of counter clear.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis

Output:

Name	Type	Description
flag	u8	0: not ready 1: cleared already and after reading, this flag will reset to 0 automatically

● lsi3123 soft homing command

Format : u32 Status = lsi3123_soft_homing_command (u8 CardID,u8 Axis)

Purpose: To do software homing (software clear counter)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis

Counter function group

● lsi3123 set counter mode

Format : u32 Status = lsi3123_set_counter_mode(u8 CardID,u8 Axis,u8 mode)

Purpose: Set counter working mode

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
mode	u8	00h: quadrature mode, A lead B upcount (default) 01h: quadrature mode, B lead A upcount 02h: single clock mode. A->Clock B->Dir and up count while Dir=1 03h: single clock mode. A->Clock B->Dir and up count while Dir=0 04h: dual clock mode. A->CW B->CCW 05h: dual clock mode. A->CCW B->CW

● lsi3123 set quadrature times

Format : u32 Status = lsi3123_set_quadrature_times(u8 CardID,u8 Axis,u8 mode)

Purpose: Set the multiple rate of quadrature input (this function is valid while counter mode is set in quadrature mode)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
mode	u8	00h: input x4 (default) 01h: input x2 02h: input x1

● lsi3123 read counter

Format : u32 Status = lsi3123_read_counter(u8 CardID,u8 Axis,i32 *value)

Purpose: Read the counter value (on the fly)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis

Output:

Name	Type	Description
value	u32	current counter value

● **lsi3123 load counter**

Format : u32 Status = lsi3123_load_counter(u8 CardID,u8 Axis,i32 value)

Purpose: Load (set) the counter value.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
value	i32	counter value to be set

● **lsi3123 latch control**

Format : u32 Status = lsi3123_latch_control(u8 CardID,u8 Channel,u8 On_Off)

Purpose: To set the external latch input source. (reference: lsi3123_latch_mode for working mode setting)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Channel	u8	0: probe_in (latch IN0) 1: foot_SW_IN (latch IN1)
On_Off	u8	0: off (disable) 1: on (enable)

● **lsi3123 latch mode**

Format : u32 Status = lsi3123_latch_mode(u8 CardID,u8 mode)

Purpose: To set the working mode of latch function. (reference: lsi3123_latch_control for external latch input source setting lsi3123_read_latched_value for external trigger latched data reading lsi3123_preset_counter for the data of external trigger to load counter)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
mode	u8	0: continuous external trigger to latch mode 1: single shot external trigger to latch mode After the latched trigger latch_control will automatically reset to OFF (disable) 2: continuous external trigger to load mode 3: single shot external trigger to load mode After the latched trigger latch_control will automatically reset to OFF (disable)

- **lsi3123 preset counter**

Format : u32 Status = lsi3123_preset_counter(u8 CardID,u8 Axis,i32 value)

Purpose: Load (set) the preset-counter. Preset-counter is used to load counter while the counter latch mode is set at external trigger to load.

(reference: lsi3123_latch_mode)

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
value	i32	counter value to be set

- **lsi3123 read latch flag**

Format : u32 Status = lsi3123_read_latch_flag(u8 CardID,u8 Channel,u8 *Flag)

Purpose: To read the latch flag

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Channel	u8	0: Probe_IN(Latch IN0) 1: Foot_SW_IN(Latch IN1)

Output:

Name	Type	Description
Flag	u8	0: no latch/load trigger occurs 1: latch/load trigger occurred

- **lsi3123 read latch value**

Format : u32 Status = lsi3123_read_latch_value(u8 CardID,u8 Axis,i32 *value)

Purpose: To read the latched value

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis

Output:

Name	Type	Description
value	i32	latched counter value

Security function group

● **lsi3123 set password**

Format : u32 Status = lsi3123_set_password(u8 CardID,u16 password[5])

Purpose: To set password and enable the security function (security_enable will set to 1 , reference: lsi3123_set_password) .

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
password	u16	the password in array format, each element in the range of 0~65535.

● **lsi3123 change password**

Format : u32 Status = lsi3123_change_password (u8 CardID, u16 Oldpassword[5] , u16 Newpassword[5])

Purpose: To change password. When security_enable=1 (reference: lsi3123_read_security_status) use this function will change old password to new one. If the old password is not correct, the security error counter will increase by 1, after 10 error trial, the card will be dead locked. If the change password is success, the security_enable will be 1 (the card security is enabled) and lock_status will be 0 (security lock is opened).

If Newpassword[5]=[0,0,0,0,0], the card will reset to security disabled state.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Oldpassword	u16	the old (previous) password in array format, each element in the range of 0~65535.
Newpassword	u16	the new password to be set in array format, each element in the range of 0~65535.

● **lsi3123 clear password**

Format : u32 Status = lsi3123_clear_password(u8 CardID,u16 Password[5])

Purpose: To clear password and disable the security function. If you try with error passwords more than 10 times, the card will be dead locked. Any further function of unlock will not response. You should send back to the maker to release the card. If clear password succeed, the security_enable will be 0 (the card security is disabled) and lock_status will be 0 (security lock is opened).

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
password	u16	the old password in array format, each element in the range of 0~65535.

● **lsi3123 read security status**

Format : u32 Status = lsi3123_read_security_status(u8 CardID,u8 *lock_status, u8 *security_enable)

Purpose: Read the security status for checking if the security is opened, locked or dead locked.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW

Output:

Name	Type	Description
lock_status	u8	0: security lock is opened, the card functions will work. 1: card is locked, the card functions will not work. 2: dead locked, the card has been open the security for more than 10 times. Any further unlock process will not response. You should send back to the maker to release the locked state.
security_enable	u8	0: the card security is disabled 1: the card security is enabled

● **lsi3123 unlock security**

Format : u32 Status = lsi3123_unlock_security(u8 CardID,u16 password[5])

Purpose: To unlock the card. Use this function to open the card function only the card's lock_status(reference: lsi3123_read_security_status) is 1(locked).

If you try with error passwords more than 10 times, the card will be dead locked. Any further function of unlock will not response. You should send back to the maker to release the card.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
password	u16	the password in array format, each element in the range of 0~65535.

Miscellaneous function group

- **lsi3123_read_parameters**

Format : u32 Status = lsi3123_read_parameters(u8 CardID,u8 Axis,u8 Parameter_NO,
u8 *Value)

Purpose: Read the parameters of any axis.

Input:

Name	Type	Description
CardID	u8	assigned by DIP (rotary) SW
Axis	u8	0: X axis 1: Y axis 2: Z axis
Parameter_NO	u8	0: HOMING_MODE 1: COUNTER_MODE 2: QUADRATURE_TIMES 3: LATCH_MODE 4: LATCH_CONTROL

Output:

Name	Type	Description
value	u8	the value set by corresponding functions

6.5 Dll list

LSI3123 Dll(V1.0) function table		
	Function Name	Descriptive Name
1	lsi3123_initial ()	Initialize lsi3123 resource
2	lsi3123_close ()	Release lsi3123 resource
3	lsi3123_info ()	Access lsi3123 physical address
4	lsi3123_set_input_polarity ()	Set input polarity
5	lsi3123_set_led_polarity ()	Set the LED polarity
6	lsi3123_read_input_polarity ()	Read input polarity
7	lsi3123_read_input_status ()	Read the input status
8	lsi3123_set_hard_homing ()	Set the working mode of clearing counter
9	lsi3123_read_hard_homing_flag ()	Read hard homing flag
10	lsi3123_soft_homing_command ()	Software homing (software clear counter)
11	lsi3123_set_counter_mode ()	Set counter working mode
12	lsi3123_set_quadrature_times ()	Set the multiple rate of quadrature input
13	lsi3123_read_counter ()	Read the counter value
14	lsi3123_load_counter ()	Set the counter value
15	lsi3123_latch_control ()	Set the external latch input source
16	lsi3123_latch_mode ()	Set the working mode of latch function
17	lsi3123_preset_counter ()	Set the preset-counter
18	lsi3123_read_latch_flag ()	Read the latch flag
19	lsi3123_read_latch_value ()	Read the latched value
20	lsi3123_set_password ()	Set password and enable the security function
21	lsi3123_change_password ()	Change password
22	lsi3123_clear_password ()	Clear password and enable the security function
23	lsi3123_unlock_security ()	To unlock the card
24	lsi3123_read_security_status ()	Read the security status
25	lsi3123_read_parameters ()	Read the parameters of any axis

7. LSI-3123 Error codes summary

7.1 LSI3123 Error codes table

Error Code	Symbolic Name	Description
0	JSDRV_NO_ERROR	No error.
2	JSDRV_INIT_ERROR	Driver initial error
3	JSDRV_UNLOCK_ERROR	Security unlock failure
4	JSDRV_LOCK_COUNTER_ERROR	Dead lock, unlock failure more than 10 times
5	SDRV_SET_SECURITY_ERROR	Password overwrite error
100	DEVICE_RW_ERROR	Device Read/Write error
101	JSDRV_NO_CARD	No LSI3123 card on the system.
102	JSDRV_DUPLICATE_ID	LSI3123 CardID duplicate error.
300	JLSLSI_ID_ERROR	Function input parameter error. CardID setting error, CardID doesn't match the DIP (rotary) SW setting
305	JLSLSI_POINT_ERROR	Input function parameter error.
306	JLSLSI_AXIS_ERROR	Axis parameter error.
308	JLSLSI_HOME_MODE_ERROR	Mode parameter error.
309	JLSLSI_POLARITY_ERROR	Polarity parameter error
310	JLSLSI_ON_OFF_ERROR	On_off parameter error
400	JLSLSI_DRIVER_NOT_SUPPORT	Driver function not support error